2016

Near real-time early cancer detection using a graphics processing unit

Jason Helms
Eastern Washington University

Near Real-time Early Cancer Detection using a Graphics Processing Unit

---

A Thesis

Presented To

Eastern Washington University

Cheney, Washington

In Partial Fulfillment of the Requirements

for the Degree

Master of Science in Computer Science

By

Jason Helms

Winter 2016

Thesis of Jason Helms Approved by

DR. YUN TIAN, GRADUATE STUDY COMMITTEE	DATE
STUART STEINER, GRADUATE STUDY COMMITTEE	DATE

Master's Thesis

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Eastern Washington University, I agree that the JFK Library shall make copies freely available for inspection. I further agree that copying of this project in whole or in part is allowable only for scholarly purposes. It is understood, however, that any copying or publication of this thesis for commercial purposes, or for financial gain, shall not be allowed without my written permission.

SIGNATURE DATE

Abstract

Automatically detecting early cancer using medical images is challenging, yet very crucial to help save millions of lives in the early stages of cancer. In this work, we improved a method that was originally developed by Yamaguchi et al. from the Saga University in Saga Japan. The original method would first decompose the endoscopic image into four color elements: red, green, blue and luminance (RGBL). Next each component is again decomposed to non-overlapping blocks of smaller images. Each smaller image undergoes two phases of DWT(s) and finally the Fractal Dimension (FD) is calculated per smaller image and abnormal regions are detectable. Our proposed method not only used GPU technology to speed up processing, this method also applied edge enhancement via Gaussian Fuzzy Edge Enhancement. After edge enhancement, multiple thresholds (or tuning variables) were identified and adjusted to reduce computational requirements, decrease false positives and increase the accuracy of detecting early cancer. Most lesions where a physician had manually indicated that could be an area of concern were detected quickly, less than four seconds, which is roughly 25x quicker than the existing work. The false positive rate was reduced but still needs improvement. In the future, a Support Vector Machine (SVM) would be an ideal solutions to reduce the false positive rate while also aiding in increasing detection and SVM technology has been implemented on the GPU. Once a technology, like a SVM, is implemented with better results, video processing will be the nearing the final step to "Near Real Time Automatic Detection of Early Esophageal Cancer from an Endoscopic Image."

Acknowledgments

Thanks are due to Dr. Yun Tian, Stuart Steiner from the Computer Science Department and Dr. Weiser from the Engineering Department the at Eastern Washington University, Cheney, WA. for the helpful guidance, suggestions, comments and providing support as advisors. Thanks to the contributors at http://grand-challenge.org/ for the inspiring ideas and also thanks are due to the contributors behind the site http://grand-challenge.org/site/endovissub-barrett/home/ for providing the test data.

Contents

List of Figures

1 Introduction

Physicians in the cancer research and treatment communities have been at battle with cancer for many years. Computer scientists have teamed up with cancer researchers to increase cancer detection results because early cancer detection is the first step towards effective cancer treatment. With the increasing quality of medical cameras, designed to collect image data from inside the human body, there is an increase in demand for image analysis. Many new methods for early cancer detection use image analysis which can be slow due to the computational load for commodity hardware or Personal Computers (PCs). Analysis of image data could reduce the duration of procedures that a patient must undergo just to collect initial endoscopic images. Image analysis could also lessen the number of screenings for a patient that has been or could be diagnosed with some form early of esophageal cancer.

1.1 Esophageal Cancer

Cancer.org reports that there were 16,980 new cases of esophageal cancer reported in 2015. Additionally, over 15,590 deaths resulting from esophageal cancer in 2015, roughly 75% men to women. This means that 1 in 125 men or 1 in 435 women will be diagnosed with some kind of esophageal cancer in their lifetime. In the late 60s and early 70s, the chance of living longer than five years after being diagnosed with esophageal cancer was roughly 5%. Today the number of surviving cancer patients diagnosed with a form of esophageal cancer is just over 20% [1]. Esophageal cancer can be broken down into 3 stages: Local, Regional and Distant. The highest survival rate belongs to Local, with a 40% survival rate. The survival rate is so high for patients with Local stages of cancer because this type of cancer is localized to one area. Most Localized staged cancer is considered resectable, meaning the cancer can be completely removed with surgery. Often when a patient has potentially resectable cancer, surgery is not an option due to the patients health.

At the Local stage a patient could battle cancer with chemotherapy, with or without

radiation, depending on the proximity to the stomach, or surgery if the tumor is smaller than 2 centimeters and sufficiently far enough away from the stomach [1]. Some other methods for early cancer treatment include: endoscopic tissue removal (some endoscopes have the equipment to remove small amounts of tissue), Cryotherapy Ablation (freezing the affected tissue off with super cooled liquid nitrogen), Radiation therapy, Esophageal Mucosal Resection (removing esophageal tissue less than 1 centimeter) or Endoscopic Submucosal Dissection (these lesions are usually between one and five centimeters) [2]. The next stage of esophageal cancer is Regional with about half the survival rate of Localized, roughly 21% [1]. The Regional stage of cancer has spread to other tissues or the lymph nodes. The last stage is Distant with the lowest survival rates. In this stage of cancer, the cancer has spread to many lymph nodes and/or other tissues, like the spine or the main artery that exits the heart.

The type of cancer targeted by the proposed method is when a Barrett's Esophagus goes undetected and/or untreated. Normally, the diaphragm would keep stomach acid from escaping the stomach and out of the esophagus because the esophageal cells are not equipped to handle the stomach acid. If the diaphragm widens, the stomach could move up and stomach acid will be able to enter the esophagus. When this happens the human body has a defense plan, the body uses small intestine or bowl cells to line the bottom of the esophagus where the esophagus meets the stomach. This is the location where stomach acid could begin to reside and sometimes this can feel like heart burn or indigestion. The small intestine cells are stomach acid resistant, unfortunately now, at the bottom of the esophagus, is a Barrett's Esophagus. When a Barrett's Esophagus goes undetected, early cancer begins to form from small lesions near the barrier between the normal esophageal cells and the small bowl or intestine cells [1]. Some patients never know that they even have Barrett's Esophagus so they do not get tested until it is too late. Then lesions could enter the later stages of cancer or the cancer could spread to other organs. The heart, lungs, liver, kidneys and the pancreas are all in close proximity. Many patients suffer from long-term gastrointestinal reflux disease (GERD). Esophagus cancer is much easier to detect when a patient is affected by GERD because there is a

better chance of having a endoscope procedure. Additionally, poor diet, smoking and being a Caucasian male can increase the chances of a Barrett's Esophagus. A diagram of the esophagus is provided in Figure 1.

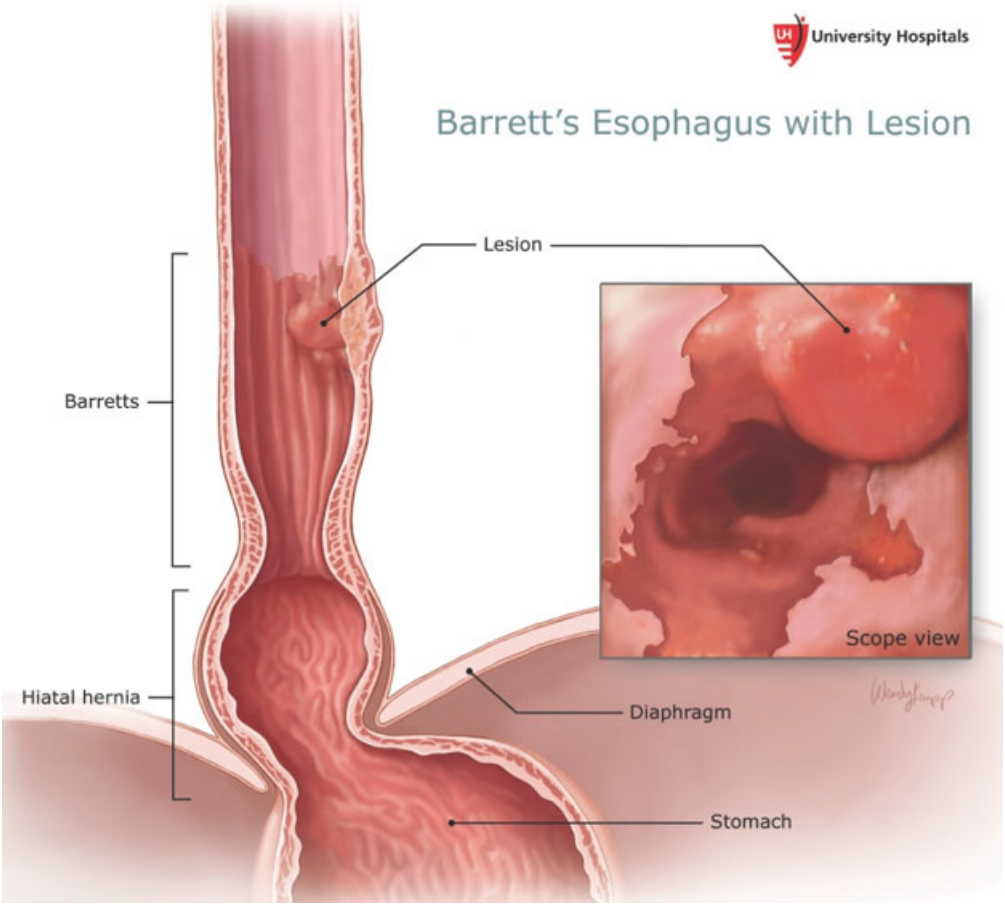


Figure 1: In this figure, the diaphragm is not doing its job and the darker red cells represent the small bowl or intestine cells that replace the esophagus cells. The body does this because esophageal cells are not acid resistant where there small intestine or bowl cell are resistant to stomach acid. The area where the darker red cells stop the normal pink cell begin is the target area for physicians. Physicians noted that targeting this area will increase early detection and reduce patient burden by reducing patient procedure time. Image referenced from http://www.uhhospitals.org, [3]

When the lesions are not caught until the later stages, patients must live with a survival rate of less than 15% but if caught early enough for the lesion, a patient has a 93% of outliving 10 years [4].

1.2 Image Analysis

Physicians have been trying to reduce the burden on the patient by targeting the lower section of the esophagus, this reduces the duration of a routine screening during the endoscopic procedure. Better imaging technology is crucial for advancements in early esophageal cancer detection. One major benefit to endoscopy technology comes from the advancement in High Definition (HD) camera technology. For example, new cell phones now have HD cameras that can take HD images and hand held video cameras shoot in 4K video. These cameras are now being used in the endoscopy technology for much higher definition imagery. The HD cameras allow for better endoscopy images, and the HD cameras reduce the time required to get endoscopy images which will further reduce patient burden. The HD cameras should also increase the detection rates of current algorithms, or help more physicians identify more lesions by simply looking at clearer images.

Advancements in HD technology in the endoscopes means that researchers will need stable and well designed libraries to perform image analysis. Thanks to the developers behind Open Computer Vision (OpenCV), researchers are getting better and faster image processing libraries and software. OpenCV is free to use, but it usually requires compiling then installing. OpenCV is a good choice for image processing because once the sequential version of code has been developed, using graphics processing speedups for the parallel conversion is as easy as compiling the image processing software then using the built in compiler support to create the parallel machine code. Some of the image processing libraries for Graphics Processing Units (GPUs) are compiled to use Compute Unified Device Architecture (CUDA) speedups and CUDA fast math. And they have been clocked at up to 30x faster than traditional Central Processing Unit (CPU) image processing libraries [5]. The CUDA enabled libraries were developed to provide developers and researchers a convenient framework compatible with the sequential CPU libraries calls, reducing the burden on the developers and researchers [5]. Finally the developers behind OpenCV decided that completeness of the CUDA integration into the

OpenCV libraries was important [5]. This means that all GPU enabled code runs on the GPU, weather the speed up results were great or not, to reduce the memory transfer time between the GPU and the CPU. If there is non-CUDA GPU on board the test computer when compiling OpenCV, the non-CUDA GPU will be used to speed up the processing, when using OpenCV because OpenCV built in GPU support for most major GPUs in 2011 [5]. The speed ups may not be the same a crossed all GPU architectures.

1.3 Image Analysis Issues

Once a physician has determined that a patient has a Barrett's Esophagus, regular screening must begin to look for lesions. Regular screenings can get expensive for the patient, and if the lesions are found in the early stages, the cancer is usually treatable [2]. Additionally, there is no guarantee that the early cancerous lesions will be caught in time by the professional using the endoscope. Often, these lesions are missed due to inexperience or the lesions are too subtle. The cells around the top of the stomach or bottom of the esophagus have already changed in appearance, making identification of early lesions more difficult. The more lesions that are caught earlier, means reducing the severity of pre-cancer and cancer treatment. However, early cancer detection using images analysis is time consuming, due to the tremendous amount of computations performed on the hardware. For example, the method proposed by Yamaguchi, et al., takes 3 minutes to process one image. In order to speed up the computation, we used a parallel platform to perform the computation on a GPU.

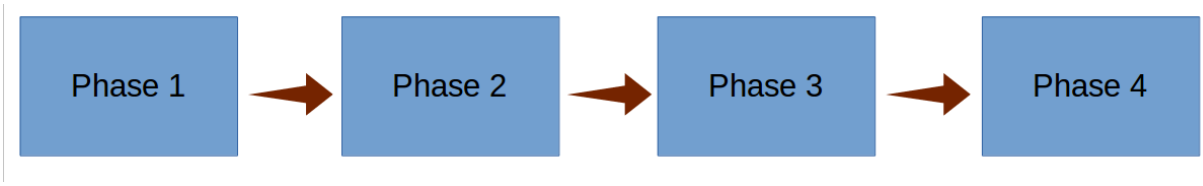


Figure 2: this is the simplified phase map. Phase 1 outputs feed into Phase 2 inputs, Phase 2 outputs feed into Phase 3 inputs, etc.

1.4 Proposed Method

The proposed method is broken into four phases discussed in section IV. The first two phases are preprocessing steps to reduce noise, and reduce the size of the input data. The first phase decomposes the image to separate color components and applies edge enhancement. The input data is 1600-by-1200 pixels but not all pixels are useful. After the image has been resized for optimal performance, the next phase reduces the input image to grid of smaller images to apply color intensity thresholding. The third phase computes multiple phases of Discrete Wavelet Transform (DWT). The final phase measures the complexity of the lesions using fractal dimension calculations via box counting.

The first two phases are not difficult but are still sometimes slow. The slowest processing comes from the box counting because box counting requires many pixel computations and may visit each pixel many times. For this reason, phase three computes two phases of DWT. This will reduce the size of the input data to a 94% smaller data set. DWT will also filter out the high frequency small veins. The formal math for the DWT is located in section 2.1 and results of the DWT are located in section 4.4

Finally, the fractal dimension is used to quantify the complexity of a pattern. A legion is a chaotic pattern, this means it is very hard to predict where or how a lesion grows. Fortunately fractal dimensions allows the chaotic pattern to be quantified with a number. This also means that once a fractal dimension has been calculated, it is not reversible, or it can be said that two different patterns can have the same fractal dimension. When the fractal dimensions are low enough, and non-zero, is an indication that the area could possible be early cancer.

Background is given in Section II followed by related works in section III. Section IV proposes an improved method for locating early signs of cancer using Discrete Wavelet Transformation and Fractal Dimension computation with some early results and demonstrative examples. Results can be located in Section V. Section VI discusses the conclusions and section VII talks about future work including potential of a Support Vector Machine.

2 Background

Physicians have a variety of options to diagnose esophageal cancer. A physician could use Lugol's Iodine, endoscopy, endoscopic ultrasonography, exploratory surgery, biopsy, Magnetic Resonance Imaging (MRI) or electrocardiogram results to look for signs of early cancer. One issue that arises in every case, is the discomfort of a patient. Patient discomfort also includes when a patient must undergo surgery, then patient is put at risk when going under anesthesia or local anesthetic. A patient testing positive for Barrett's Esophagus must undergo regular screenings to watch for cancer. To avoid as much patient discomfort as possible with still providing accurate results makes image analysis a very useful tool for early cancer detection. The test images were obtained from a sub-challenge of the Endoscopic Vision Challenge from a community called "Grand Challenge in Biomedical Image Analysis", called "Early Barrette's Cancer Detections Sub-challenge" [4]. Researchers competing in the biomedical image analysis competitions come together to test proposed methods, compare notes and improve on cancer detection algorithms. The testing image set came with one set of training images, and another set of images that were the expert physicians annotations, meaning the annotation was given in a separate image with a similar name. Black portions of the doctors annotation are areas determined not to have signs of early cancer, white portions of the physicians annotation are areas of possible early cancer. Next each one of the images undergoes two rounds of Discreet Wavelet Transforms to filter noise out of the color spectrum patterns. These patterns are not human recognizable but they could be detected with Fractal Dimension Decomposition.

2.1 Barrett's Esophagus

The type of cancer targeted by the proposed method is when a Barrett's Esophagus goes undetected and/or untreated. Normally, the diaphragm would keep stomach acid from escaping the stomach and out of the esophagus because the esophageal cells are not equipped to handle the stomach acid. If the diaphragm widens, the stomach could move up and stomach acid will be able to enter the esophagus. When this happens the human body has a defense plan, the body uses small intestine or bowl cells to line the bottom of the esophagus where the esophagus meets the stomach. This is the location where stomach acid could begin to reside and sometimes this can feel like heart burn or indigestion. The small intestine cells are stomach acid resistant, unfortunately now, at the bottom of the esophagus, is a Barrett's Esophagus. When a Barrett's Esophagus goes undetected, early cancer begins to form and turns to late stages of cancer. Some patients never know that they even have Barrett's Esophagus so they do not get tested until it is too late. Then lesions could enter the later stages of cancer or the cancer could spread to other organs. The heart, lungs, liver, kidneys and the pancreas are all in close proximity. Many patients suffer from long-term gastrointestinal reflux disease (GERD). Esophagus cancer is much easier to detect when a patient is affected by GERD because there is a better chance of having a endoscope procedure. Additionally, poor diet, smoking and being a Caucasian male can increase the chances of a Barrett's Esophagus. A diagram of the esophagus is provided in Figure 1.

2.2 Discrete Wavelet Transform

One issue the Discrete Wavelet Transform mitigated was reducing the noise of small veins from the input image. These small veins can make intricate patterns that may be detectable by box counting. Figure 20 shows the original color component image, for this example the green component is shown. The green image in figure 20 is a 3 channel image, the second image on the first row is the green image converted to gray scale. The second row in figure 20 is magnifying the south west region of the input image in gray scale, and the third row is the magnification of a 128-by-128 pixel block. The next

two rows in figure 20 are the first phase of DWT and the second phase of the DWT. The last row in figure 20 is the comparison of the 128-by-128 input block from line 3 and the output after two rounds of DWT. The veins are not completely gone but they are filtered or blurred enough that they will not affect box counting as much as before DWT. Anther added benefit of the DWT is the input reduction. The green image is 1024-by-1024 pixels, over a million pixels in total. After the second phase of applying the DWT, the output image would only be 256-by-256 pixels, about 65K pixels. This is almost a 94% reduction of the input data, plus the vein filtering makes DWT a powerful tool.

The Fourier Transform (FT) needs some introduction before DWT is explained. The FT was developed by Joseph Fourier in the 1700s [6]. The FT is able to analyze a signal by creating mathematical structures that vary in scale [6]. This allows the FT to detect the frequencies in a signal. The FT may be able to detect the frequencies in a signal, but it cannot tell us what time those frequencies occurred, see figure 3 for a visual example.

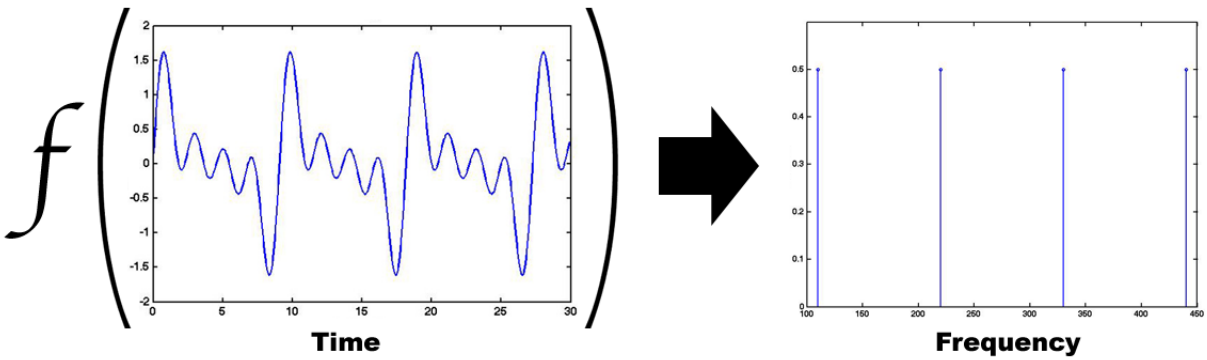


Figure 3: This is a general example of fourier transform frequency detection results, image obtained from http://devonbryant.github.io, [7]. The y-axis for both graphs represents the amplitude of the frequencies. This example also shows why the DWT is used over FFT. The time resolution for the FFT results is non existent.

This is great for a stationary signal but when the signal is not stationary, like figure 3, the FT indicates nothing about when the frequencies occurred. There is a solution to get some information about when the signal occurred and this solution is to use the Short Time Fourier Transform (STFT). The STFT will apply a time window to the signal to get some frequency information for periods of time in the original signal [8]. The STFT

will assume the signal in the window is stationary which may not be the case, most of the real world signals are not stationary in time. The STFT window will give poor frequency resolution with a short or narrow window and poor time resolution with a long or wide window of time [8].

Additionally, the Heisenberg principal says that the time the frequency occurs is still unknown [9]. The solution to the the Heisenberg principal is Multi-Resolution Analysis (MRA), which will lead to DWT very soon. MRA will analyze the signal at different frequencies with different resolutions. One direct advantage of Wavelet Transforms (WTs) have over FT is that the width of the window is changed as the WT is computed for every spectral component [8]. A WT will split an input signal into many signals where each signal corresponds to a different frequency band [8]. WT will also provide the frequencies that occurred in the signal at specific time intervals, something the FT cannot do. Two types of WT are explained next, the Continuous Wavelet Transform (CWT) and the WT used in the proposed method the Discrete Wavelet Transform.

The CWT is a WT that computes a Mother Wavelet, this wavelet is the base wavelet for prototyping and generating the other wavelets to approximate the input signal [8]. While the window is of finite length, the window is compressed, or dilated and shifted to obtain the new wavelets [8]. This also means that a CWT is not a true DWT. The DWT series is simply a sampled version of the CWT, and the information provided is redundant [8]. On the other hand, the DWT provides enough information for both analysis and synthesis [8]. The DWT will reduce the computation time sufficiently, it is easier to implement and most importantly a DWT will decompose the signal into a coarse approximation of the input signal with detail information [8]. This means that one of the outputs of the DWT is a smaller approximation of the input signal, which could reduce processing. The DWT is able to detect good time resolution with poor frequency resolution at high frequencies [9]. The DWT is also able to detect good frequency resolution with poor time resolution at low signals [9].

DWTs gained popularity some years ago for data compression. DWT can be also classified using Daubechies Wavelets. The simplest of Daubechies Wavelets is the Haar

Wavelet. The Haar Wavelet was first discovered by Hungarian Mathematician Alfred Haar in 1909 [10]. The Haar Wavelet is the simplest to compute, mainly consisting of additions and subtractions. With image processing, there are four smaller outputs generically named: HL, HH, LH, and LL. LL is the sub-sampled image of the input image or the approximation of the image, see section 4.4 for examples. HH, HL, and LH are described as the horizontal, vertical and diagonal details of the input image. The formal math equation is provided in figure 4 and some implementation demonstrations and results can be found in Section 4.4.

$$LL(i,j)=\sum_{n=0}^{2M-1}\sum_{m=0}^{2M-1}(p_m p_n \cdot I(m+2i,n+2j))$$
$$LH(i,j)=\sum_{n=0}^{2M-1}\sum_{m=0}^{2M-1}(p_m q_n \cdot I(m+2i,n+2j))$$
$$HL(i,j)=\sum_{n=0}^{2M-1}\sum_{m=0}^{2M-1}(q_m p_n \cdot I(m+2i,n+2j))$$
$$HH(i,j)=\sum_{n=0}^{2M-1}\sum_{m=0}^{2M-1}(q_m q_n \cdot I(m+2i,n+2j))$$

$$p=p_n \quad and \quad q=q_n(n\in\{0\ldots 2M-1\})$$
$$q_n=(-1)^n p_{2M-1-n}$$

Figure 4: This is the formal definition of the DWT processing.

In practice, first the rows of an image are computed then the columns of the image, which means the calculations are performed with two passes over the input image using sequential processing. Each pixel is convoluted with a filter. All input pixels are visited on the first pass and half the number are visited the second pass. Not all pixels could be processed at the same time in sequential processing. However, a low cost desktop

GPU, like the NVIDIA brand, typically processes a 32-by-32 pixel block at one time. Additionally, there would be no need to first process the rows and then the columns on a GPU. All pixels could be processed in far fewer passes over the input image when computing 1000 pixels at the same time on a GPU.

2.3 Fractal Dimension

Fractal Dimension (FD) is used to measure complex patterns by characterizing patterns when measured at different scales. Some patterns, usually produced by nature, are extremely hard to predict but the complexity of these patterns is easier to detect using FD. These patterns can seem chaotic, but this is acceptable for FD. It is accepted that a system can be anything that has more than one part and the system is dynamical when the systems states change with time [11]. The system is also said to show nonlinear dynamics when one action from one element to another element is not proportional to the action itself [11]. Simple systems that show nonlinear dynamics often generate random-like effects, or chaos [11]. This could also describe how a lesion grows, meaning it is known that they spread and have a starting point. How the lesion spreads is unknown, or chaotic. Fortunately there is something predictable about a chaotic growing pattern, it is not completely random. A fractal object has a property that states that more of the fine structure, or smaller details, are revealed as the object is magnified [11].

Like the Koch snowflake, a lesion is structured in a similar fashion. As the Koch snowflake is magnified, it can been seen that the pattern changes at every level. The same is not true for natural patterns. Naturally reoccurring patterns, like the growth pattern of esophageal cancer, are self-similar at different levels of magnification [11]. FD also allows us to measure patterns that take on a non-integer dimension. Integer dimensions are exponents of measurements, i.e. if S is a side of a square, then area of a square is S^2 and the dimension is two, the same as the exponent. This means that when a point turns into a line or when a line turns into a plane, those points are moving from a one dimensional space to a two dimensional space or a two dimensional space to a three dimensional space respectively. A FD that takes on a non-integer values indicates

that the FD is moving from one dimension to another, which comes in handy when early cancer lesion begin to grow.

The method used to compute the fractal dimensions in their digital form is the box counting method because of the elegance of the implementation. Box counting is like putting a piece of graph paper, with grid size H, over the pattern and then count the number of grids that contain the pattern and call this P [11]. The formula for fractal dimension is D = $log(P)/log(1/H) = -log(P)/log(H)$. As the size of H decreases and the complexity of that pattern increases, means that there will be more grids filled with the pattern. The box counting method is easy to implement, it used to be very sequential in nature yet adaptable to parallel processing. The minimum detection accuracy for fractal dimension is 64-by-64 pixels [13]. To see examples, diagrams and results of box counting, see section 4.5.

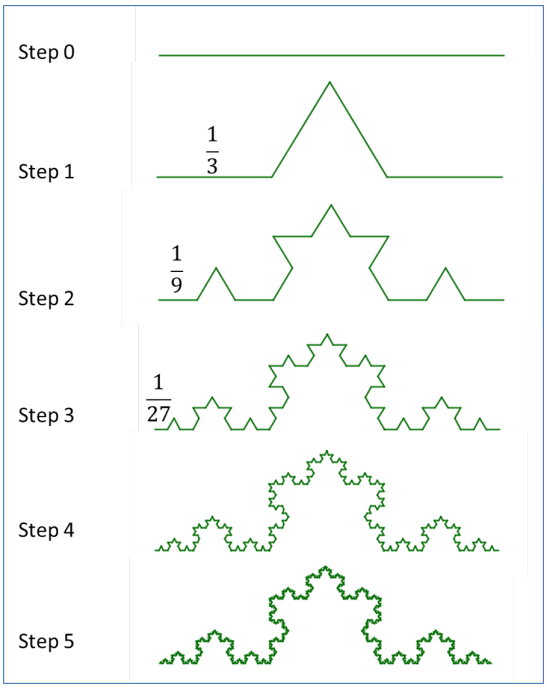


Figure 5: This is an example of the steps in building a Koch snowflake, image obtained from https://danpearcymaths.files.wordpress.com, [12].

Figure 6: This is one of the famous examples of fractal dimensions. In this images, it can been seen that when the length of the measuring stick changes, 200km, 100km, 50km, the length of the coast line changes. This was the essential idea behind box counting, the algorithm used to measure the fractal dimensions in phase 4. Image obtained from http://soft-matter.seas.harvard.edu, [14].

2.4 Open Source Libraries

There is a benefit to use free open source software. Some commercial software comes with a free version or trial application. However, the functionality is greatly reduced. Open source software and libraries come with full functionality. Also, most open source software is free to alter and use or sell at a developers discretion. One important element to open source or collaboration is communication. The open source communities usually use forums for open communication, like stackexchange.com for example, to open channels of communication for anyone interested. There is also a large on line community for OpenCV, C++ and almost any other academic branch of science.

Innovation and advancement in technology comes from the collaboration of individuals and teams. However, some major corporations are now opening up their independent languages to the open source communities, like Swift from Apple. Open source software and libraries let developers and researchers experiment without major funding while enhancing collaborate efforts to better technology.

The use of open source libraries also requires additional research or testing which

could be more extensive than commercial software testing. Most commercial software comes pretested for what the company assumes the software will be used. Additionally, commercial software is not readily altered and does not come with the source code, where open source software can be altered at the source code level to do what a developer needs.

Open Source Computer Vision (OpenCV) is a set of open source libraries used for image processing. The image processing was developed using OpenCV with C++. The phases were managed using QT Creator, which is C++ compatible. The OpenCV libraries are actually maintained by a company called Itseez [16]. These libraries were also used for the automatic parallel conversion because OpenCV has built in GPU libraries for the major architectures. These GPU libraries were added to OpenCV in 2011, with version 2.4, and these GPU libraries can used nearly any GPU attached to a test system. The OpenCV libraries contain the functionality to perform the image processing without changing code between recompiling the libraries to use different GPU speed up libraries. Both OpenCV and QT creator have C++ compatible libraries, making the processing of using OpenCV with QT Creator very efficient.

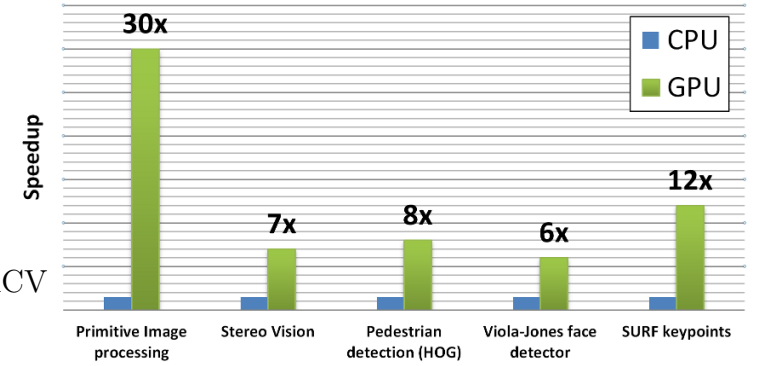


Figure 7: This is a graph of average speed up results using the OpenCV libraries compiled to use CUDA. Care must be taken to use library calls and to set up a matrix in the same way as the standard OpenCV demonstrations [15]. Image obtained from http://opencv.org, [15].

2.5 Graphics Processing Units

In the last decade, Graphics Processing Unit (GPU) technology has been used to speed up general and scientific processing. For instance, the GPU allows Physicists and Movie Makers the ability to trace light rays for science and movies. A GPU is a special Central

Processing Unit (CPU) designed for graphics processing, but in recent years, researchers have realized the values of parallel processing on the GPU.

2.5.1 Intel

Intel has began developing integrated graphics chips called Iris Graphics in 2014-2016 [17]. These graphic chips are built into the Central Processing Unit, which has pros and cons. The number one pro is that the CPU can read the Level 2 cache of the graphics processor. The con is that the graphics processor has to share the Random Access Memory (RAM) with the CPU.

Intel has another major graphics processor family called Intel HD Graphics [17]. These graphics processors are built on the same CPU architecture as the commodity desktop hardware. This also means that these graphic processors must share the RAM with the CPU. The Intel graphics processors are not widely used for GPU speedups, but they are great for low cost on board graphics processing and testing on a budget.

2.5.2 NVIDIA

A NVIDIA GPU is made up of many smaller CPU cores called stream processors, or CUDA cores [18]. Each core has similar elements of a standard CPU [18]. This means that each core has local memory and a local cache to process data. More importantly each core can handle one thread, or one unit of processing. For any processing that is done on a pixel by pixel basis, CUDA can be used to process all the pixel computation in parallel or at the nearly the same time. The claim is nearly the same time because low end NVIDIA GPUs execute a block of 32-by-32 threads at one time, this block of threads is called a warp [19]. The other warps, if any, wait in a queue until a block CUDA cores are ready to process them [19]. So the processing is not 100% parallel, meaning processing every pixel at the same time is not possible but processing over 1000 pixels at one time is what actually happens. This will still speed up image processing by many magnitudes.

2.5.3 Advanced Micro Devices

Advanced Micro Devices (AMD) is another major brand of graphics processors. These graphics processors work like the Intel Graphic Processors. They are built using the same CPU technology that AMD uses for general CPUs but these graphics processors come with their own Random Access Memory (RAM) or dedicated memory [20]. Recently, AMD has started developing Graphics Processing Units (GPUs) with multiple cores to decrease computation time [20].

3 Related Work

In this section, we talk about some existing related research work. Their work was not an end all to a humans manual cancer detection. But their work, like the proposed method is supposed to assist the physician and technicians conducting endoscopy procedures. All work described in this section and the proposed method also intend to reduce the patient burden by reducing the number and length regular screening of the patients. Additionally, all work described in this section and the proposed method aim to increase the number of early detected lesions which will in turn reduce the number of stage 3 patients and the mortality rate of lesions detected too late.

All of the methods could better detect early signs of esophageal cancer if working together than they could all on their own if a SVM is used to obtain optimal thresholds. The collaboration of methods would also optimize the time a physician needs to determine if a patient has early signs of cancer. When the application of a Support Vector Machine is complete, the results could help train new physician to better detect lesions as well from the proposed method alone.

3.1 Automatic detection of early esophageal cancer from endoscope image using fractal dimension and discrete wavelet transform.

The proposed method is based on the work done by the researchers Yamaguchi et al. from the Saga University in Saga Japan working on “Automatic Detection of early esophageal cancer from endoscope image using fractal dimension and discrete wavelet transform.”

They were able to detect acceptable results for ten images from four patients. This work resulted in fairly slow processing performance but promising results. Additionally the hardware used in their testing was an older slower desktop computer. It could take up to three minutes to determine a result of positive or negative early cancer signs for each image [13]. Some of the issues the researchers had to mitigate were that the images used were much smaller with a low resolution. This meant that the algorithm developed by the researchers needed to mirror pixels, crop out text and unusable pixel data [13]. Our proposed method used an image with the dimensions 1200-by-1600 due to new endoscopes getting better cameras, some even claim to be High Definition (HD).

3.2 Supportive automatic annotation of early esophageal cancer using local gabor and color features

The researchers Fons, et al, used the color variation with texture features as well as the a Gabor filtered image's generated from the input image. These calculations were used as vector inputs for a Support Vector Machine (SVM)[21]. A SVM is a supervised learning data mining artificial intelligence models to classify data into two groups based on training sets. With this method, this group of four used images from 7 patients, who had developed 38 lesions and were able to detect 36 of these lesions with a 75% accuracy [21]. This work concludes that the integration of new endoscope and HD video technology is bringing about new advances in cancer detection.

3.3 Analysis System of Endoscopic Image of Early Gastric Cancer

The researchers Kim G., Kim K. and Kim S. working in Korea are also using endoscope images to determine if lesions show early signs of esophageal cancer. This work starts by calculating an Index of Hemoglobin (IHb) and then the color changes in the image to detect lesions in patients. This method by Kim, et al also will provide texture data on areas indicated by the detector (physician in most cases). Kim, et al, also noted that the early detection is easily missed but the chances of surviving the later stages in Korea are less than what the US and other countries can achieve, which is why early detection is so important. In Korea, if a lesion goes undetected and enters the last stage of esophageal

cancer, the patient has less than 0.12% chance of surviving 5 years [22].

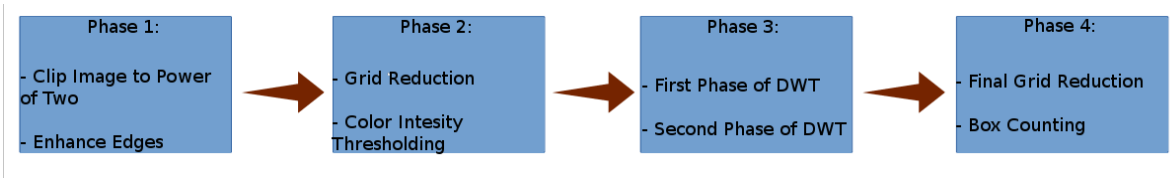


Figure 8: this is a phase mapping of the proposed method.

4 Methodology

The process of early cancer detection for Barrett's Syndrome is determined using an endoscopic image taken of the esophagus at the area where the stomach meets the esophagus. This image is taken via an endoscope during one, of usually many, procedures to check for early cancer. The results will indicate if that image contains traces of early cancer, usually with some percentage of false positives. False positives are not of high concern because missing a true positive is much more detrimental to the patient, but lower false positives means the physician can optimize their time. First the image is decomposed to the red, green, blue and luminance color components contained in separate image data structures and the size is reduced to a power of two for faster processing during the last phases. Each color is decomposed to a grid of non-overlapping smaller images. Next each smaller image (block) undergoes two passes of Discreet Wavelet Transformations (DWTs), then the blocks are further decomposed to four smaller blocks, making up a small non-overlapping grid. In this phase of processing, we compute the Fractal Dimension (FD) of each block. The Box Counting algorithm was used for the FD computation and a FD value is computed for every block. The results for corresponding block for each color are multiplied together to get a grid of 16-by-16 grid of FD values. Each FD value maps to a 64-by-64 block of pixels in the original image. If this sum is small enough and non zero, then this area is a candidate for early cancer.

To improve the results of the work just described, each image undergoes edge enhancement to enhance the low frequency edges. Gaussian Fuzzy Edge Enhancement is used to enhance the edges which means better box counting results. The edge enhancement

is applied to each color component (RGBL) before any Discrete Wavelet Transform (DWT) processing. The Gaussian Fuzzy Edge Enhancement technique involves computing a fuzzy image from the original using a Gaussian Blur function built into OpenCV. Then the enhance image is built by performing a weighted addition of the two images, fuzzy and the original, which breaks down to matrix addition and scalar multiplication.

The next obstacle to better results and early cancer detections of the identification and modification of different thresholds. The first threshold identified was the color intensity threshold. This threshold is explained in section 4.3, Pre-wavelet processing.

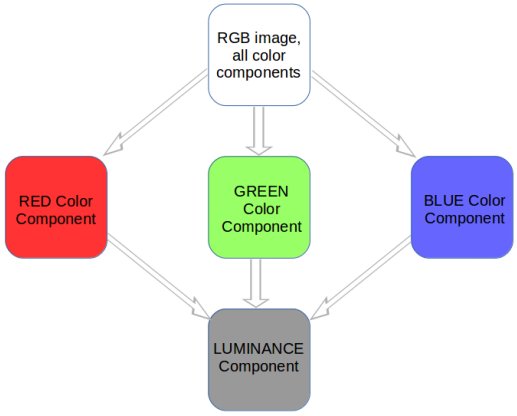


Figure 9: this is a generic example of the RGB decomposition and luminance calculation.

The next threshold holding back results was the dynamic threshold value when converting an image block to binary. This value was, again, first set at the blind average. This too proved to be too high, and after experimentation, reducing this threshold by 20% improved box counting results. An explanation of the dynamic threshold, how it was applied, is explained in section 4.5, Box Counting.

The processing in broken into four phases. Phases were chosen over a simple list of steps due to the changes of thresholds at different processing stages. The first phase will decompose the image to the separate (RGBL) color components and clip the image to a size of 2^n-by-2^n. The second phase will decompose each color component image into a grid of smaller images and apply color intensity thresholding. The third phase applies two rounds of DWT for the input reduction. The final phase is the Box Counting phase. From this point, some elegant statistics are used to determine areas of possible early cancer.

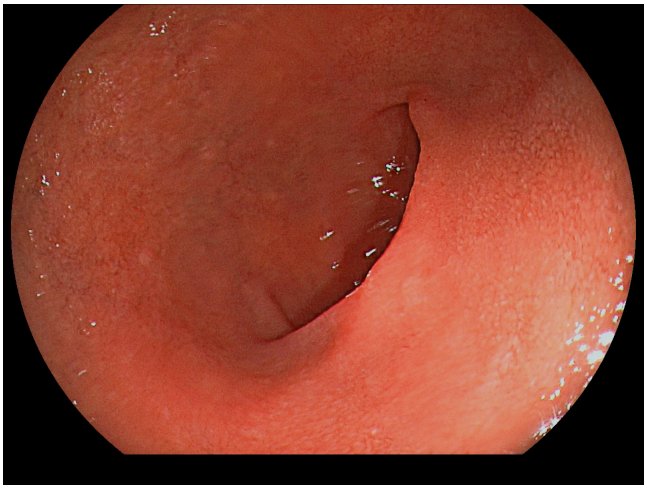


Figure 10: Patient 1, image 1. A sample patient endoscopic image. Expert physicians concluded there is no signs of early cancer in this image. This is an example of the data used, some examples use other images for processing clarity, they are just examples.

4.1 Phase 1: Clipping

All the images used in testing were 1600-by-1200 pixel, three color channel images. An example image can be seen in figure 10, this a an image that expert physicians that determined there were no signs of early cancer. First the input image is decomposed to: red, green and blue images. This marks the point where all images can be saved as a single channel image and treated like a gray scale image. Next the fourth color component, Luminance, is computed from the red, green and blue color components. Based on the visible color spectrum, Luminance is a weighted average of each color component.

Sample Luminance Calculation

1) Let Y represent the Luminance value for any pixel.

2) Let R be the red color component intensity, G be the green color component intensity and B be the blue color component intensity.

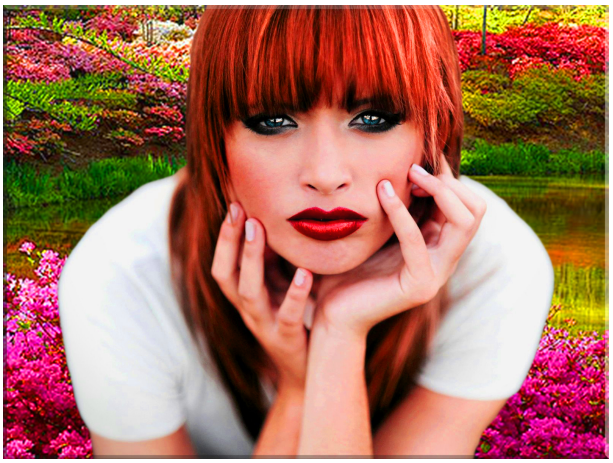
3) Then $Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$.

Next each color component image is resized from 1600-by-1200 pixels to 1024-by-1024 pixels. This is done by first finding the center of the input image, then measure 512 pixel in the north, south, east and west. These points mark the middle of the box sides of the image to be cut out, i.e. the middle 1024-by-1024 pixel block. Now the edge enhancement

is applied to sharpen or enhance edges. After all processing from phase one is complete, for each input image there will be four output images of size 1024-by-1024 pixels.

4.2 Phase 1.5: Edge Enhancement

Every color component underwent edge enhancement with Gaussian Fuzzy Edge Enhancement. The luminance color component is shown in figure 12(a), figure 12(b) is the color component converted to gray scale before edge enhancement and figure 12(c) is the results after Gaussian Fuzzy Edge Enhancement. The Gaussian Fuzzy Edge Enhancement method involves calculating a fuzzy image of the original using a Gaussian Blur function built into OpenCV, then performing a weighted sum of the two images. The weights for the sum were: 1.8 for the original input image and -0.5 for the fuzzy image. The weights for the sum actually increased most areas without affecting the image negatively. Almost like built in brightness increases, or two benefits from one computation, just like DWT.



(a) Sample 1600-by-1200 image

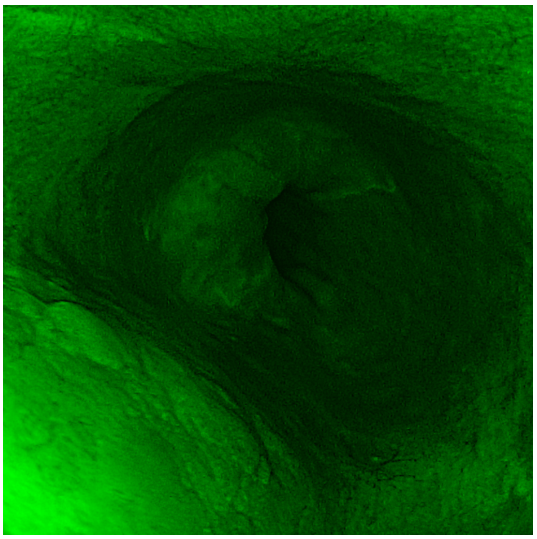
(b) Sample Red Component

(c) Sample Green Component

(d) Sample Blue Component

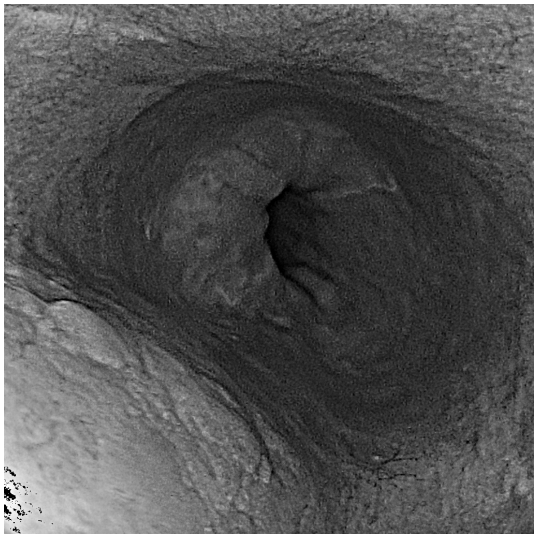
(e) Sample Luminance Component

Figure 11: The images of single color components have been saved as triple channel images for clarity of the data, from this point on, all images are displayed as gray scale, but actually represent the color intensity. Example image obtained from [23].



(a) Luminance component before gray scale conversion

(b) Luminance component after converting to gray scale



(c) The Luminance component after edge enhancement

Figure 12: This is a demonstration of the edge enhancement. A test image was used for clarity.

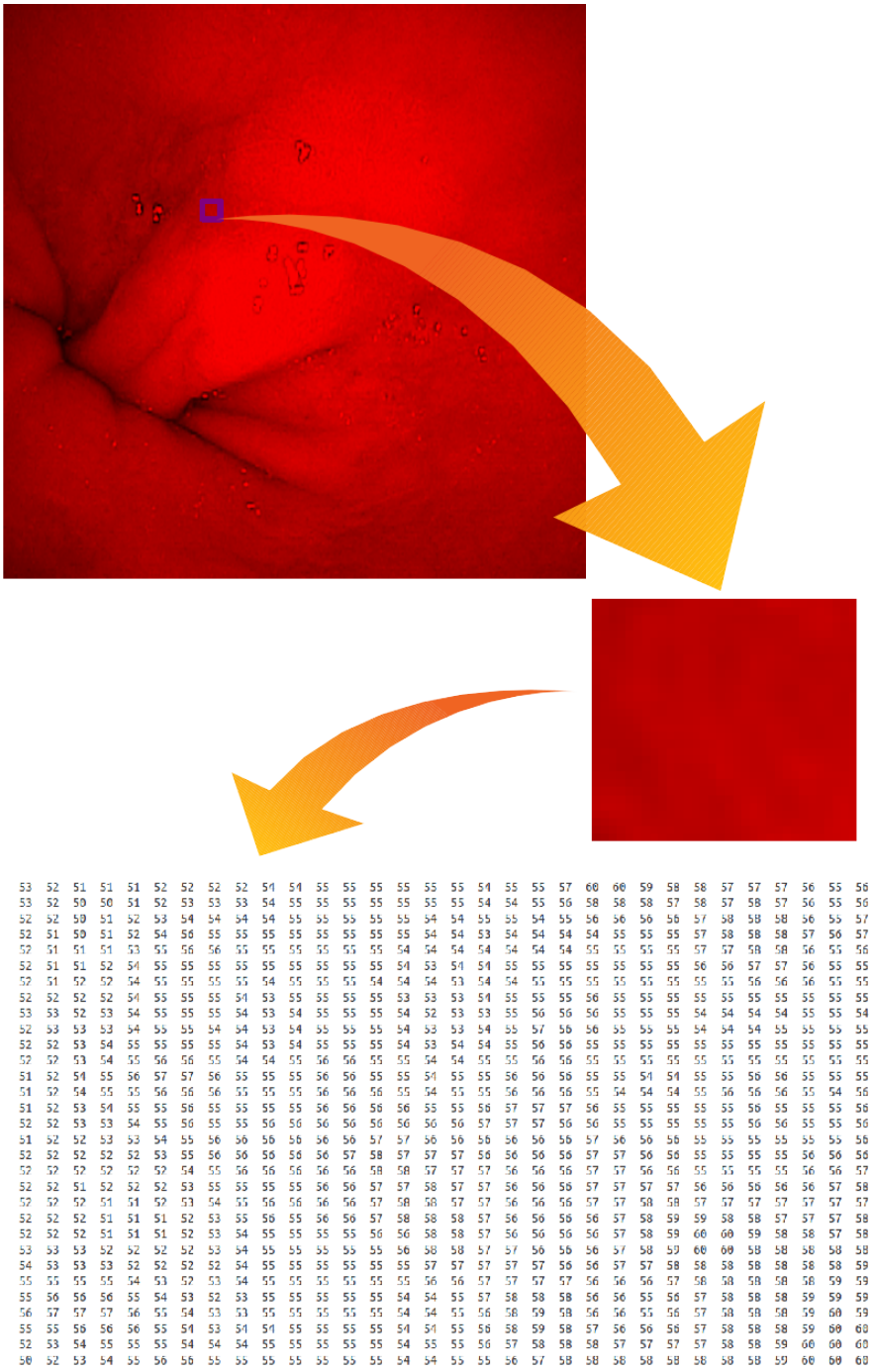


Figure 13: The red component is hard to work with because the red component has values that are so low with very little variance. The pixel values for the purple box are provided to show how bad the data is, meaning no variation and the values themselves are very low.

4.3 Phase 2: Pre-wave Processing

The input images to this phase are 1024-by-1024 pixels, single channel images. The output images will be 128-by-128 pixel blocks that make up a grid of the input image

which means there will be sixty-four output images for every color component (RGBL). Before the image is divided into smaller pieces, the average color intensity is calculated and stored as a global image color intensity.

When the color intensity threshold is set to a simple (or blind) average, there are many occasions where the areas mark by physicians annotations, were blacked out before any chance of detection. After experimentation, it was found that the blind average was too high. Relaxing this threshold by 20% still eliminated blocks with average color intensity too low to produce results without affecting areas of possible early cancer. After further experimentation, the average color intensity for the entire image was reduce by 40% to only zero out a few blocks, leaving the rest for further processing. Assuming the original intent of color intensity thresholding was to reduce the amount of processing in phase 4, color intensity thresholding with the intent on reducing processing time is of no use with GPU technology. Almost all pixels are processed at the same time, so the color intensity thresholding is used only to reduce areas of the images that may be too dark to analyze with box counting.

The color intensity thresholding is applied in the simple steps that follow:

Step 1: If the average pixel intensity for the whole image is I then $I = \alpha \cdot I$ where α is less than 1.0, see figure 14 for a general example.

Step 2: The local color intensity is calculated per block, call this 'i'.

Step 3: If the local color intensity, i, is less than the global color intensity threshold for block, j, then block j is set to zeros.

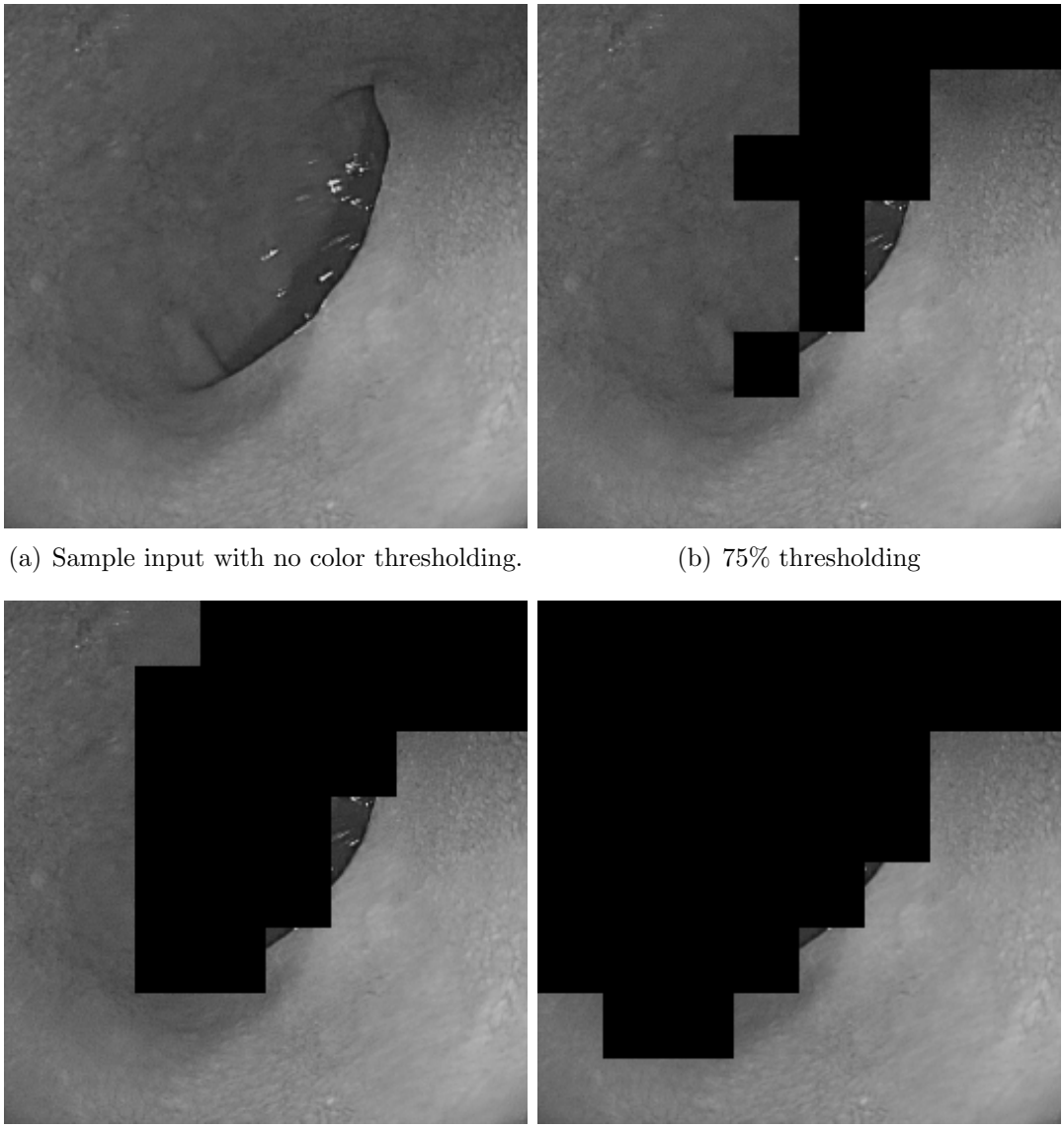
Step 4: Repeat steps 2 and 3 for every block of every color component.

$$T_{initial}: \text{ is the blind average threshold}$$
$$T_{final}: \text{ is the final/calculated threshold}$$
$$\alpha \in (0,1]$$
$$\text{Calculated final/threshold is}: T_{final} = \alpha * T_{initial}$$

Figure 14: this is a generic example of the application of thresholds to the blind thresholds, most blind threshold are set too high. See figures 16(a), 16(b) and 16(c) for a demonstrative example.

Some blocks will be set to all zeros because the color intensity values are too low and there is not enough variation between pixels. When the values are low and close enough together, or nearly equal, edges are much more difficult to detect, and color changes are much harder to detect.

Another rational of this step is that lesion areas correspond to brighter areas in the image, areas brighter than the average color intensity. Figures 15(a), 15(b), 15(c) and 15(d) show some results of relaxing the color intensity threshold at different values. Compared to [13], the best color intensity threshold is between 60-75%. The smaller the threshold (T) is relaxed, the more input data get processed in later phases. A large T value is good for a reduction in processing time but could be detrimental in finding an results, see figures 16(a)-16(c) as example of what happens when the global color intensity threshold T is too large. Obviously, if the global color intensity is set too high, none of the input data will be processed later, crippling our proposed method before it could start.



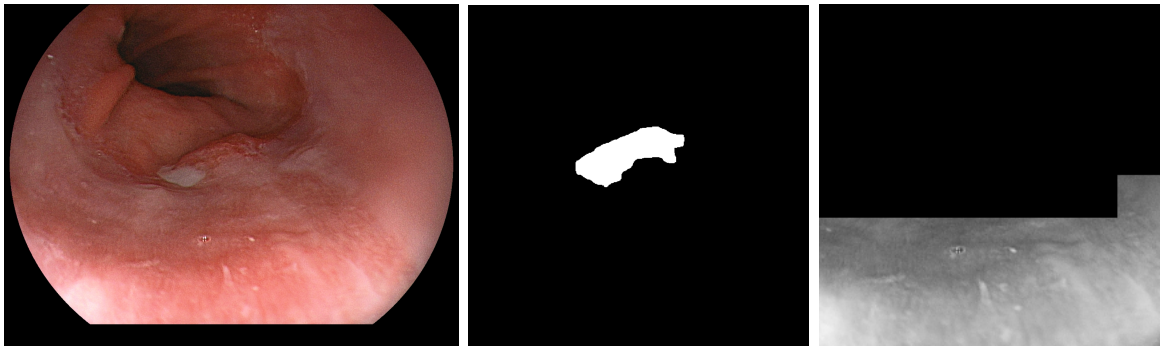
(a) Sample input with no color thresholding.

(b) 75% thresholding

(c) 80% thresholding

(d) blind average

Figure 15: The image on the upper left is all the blocks from phase two stitched together to show the input image for no color thresholding, the image on the upper right is results from 75% of the average global color intensity. The image on the bottom left are the results from 80% of the average global color intensity, and the image on the bottom right is the result of blind average thresholding, greater than 100% thresholding would leave more than half of the blocks set to all zeros, crippling the potential lesion areas to be found.



(a) Patient 19, image 1, 1600-by-1200 pixels

(b) Physician annotation of suspected areas of early cancer, Patient 19, image 1 1

(c) Crippling color intensity threshold, Patient 19, image 1

Figure 16: This is what happens when the color intensity threshold is set to a blind average. The proposed method has no chances of finding the area recommended by physicians with this color intensity threshold.

4.4 Phase 3: Discrete Wavelet Transform

The input image(s) of each color component to this phase are 1024-by-1024 pixel images. This means that for each color image, there are over one million pixels to check for early signs of cancer before phase three is applied. This can be time consuming, even for a GPU running many pixel operations in parallel, because there is still physical limitations on the number of threads running at one time on a GPU. Performing two phases of DWT on sixty-four 128-by-128 pixel blocks making up a grid of the input image will reduce those blocks to 32-by-32 pixels, reducing the size of the original input image to 256-by-256 pixels. The two phases of DWT reduce the image size by 94%, reducing the number of pixels for each color component to less than 66K pixels. This is done by the computing LL of the DWT for both rounds. None of the other outputs for DWT are computed, LH, HL and HH as they are too noisy.

Figure 17: This is one example image before the application of the Discrete Wavelet Transform. 1024-by-1024 single channel image. Figures 18(a) - 18(d) shows the results from a full Haar's wavelet transform, all four output images are shown but only LL is kept in the proposed method. This images is not a testing image, image is shown for demonstrative purposes. Example image obtained from [23].

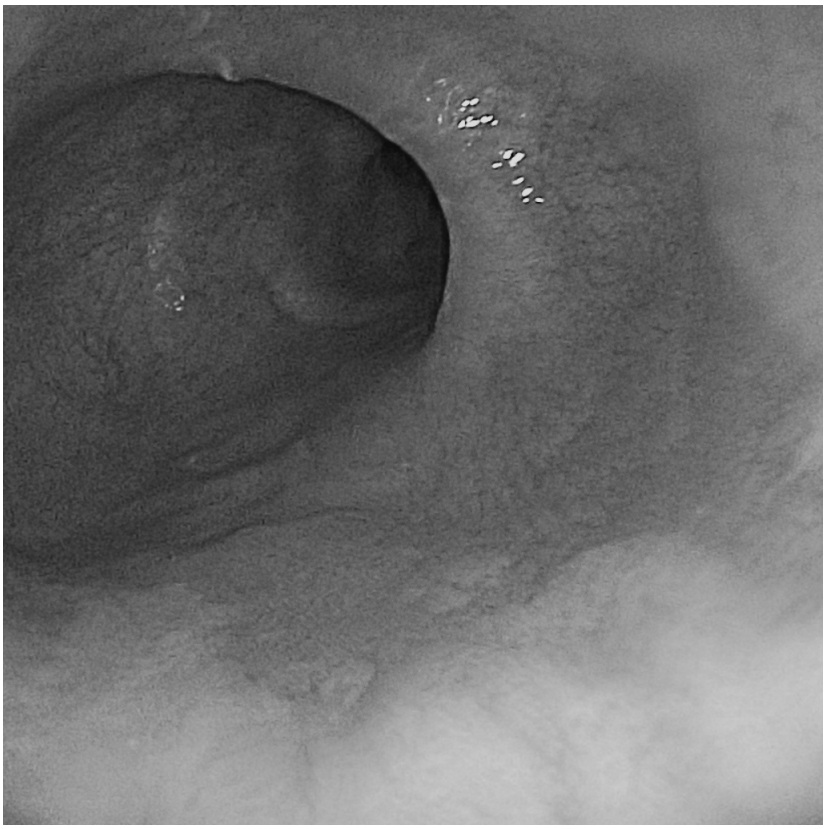
(a) Sample HL, currently 512-by-512 pixels.

(b) Sample HH, same dimensions as HH.

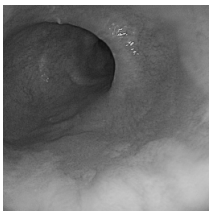
(c) Sample LL, currently 512-by-512 pixels.

(d) Sample LH, same dimensions as LL.

Figure 18: These are the LL, LH, HH and LH outputs from one full round of DWT on figure 17, meaning the rows and the columns were processed. The size of each smaller image is 25% of the original input image, figure 17. The generated images were obtained from performing the DWT on the image obtained from [23]



(a) Sample DWT input, all blocks displayed as one image.



(b) Sample DWT output, all blocks displayed as one image.

Figure 19: The image on the left is all the blocks from phase 2 stitched together to show the input image, the image on the right is the output blocks from phase 3 stitched together to show the input image. If the larger image was actually 1024-by-1024, then the smaller image would be 256-by-256, images are shown for scale. Also note that this color component is the luminance component, the other color components have been omitted because the results of the input reduction are the same but harder to see.

Finally, the results from the small, high frequency vein filtering is displayed in figure 20. In figure 20, the top left image is the green color component and converted to gray scale for processing. The second row in figure 20 is the magnification of the northwest region of the gray scale. The third row is further magnification of a 128-by-128 pixel block. This block goes through two phases of Discrete Wavelet Transform (DWT). The last row in figure 20 is the comparison of the 128-by-128 pixel block before two phases of DWT and the output of two phases of DWT. The small, high frequency veins have been filtered enough to let the box counting method identify early cancer areas without these areas triggering a false positive.

4.5 Phase 4: Box Counting

Each image block is now 32-by-32 pixels and will go through one last computation phase. Each block is split into four non overlapping images making up a 2-by-2 grid of the input block. Next each image block is converted from a single color component to binary. Converting each image to binary is done via Dynamic thresholding. Dynamic thresholding is where a box of pixels, centered at $[2 \cdot R + 1]$-by-$[2 \cdot R + 1]$, the pixel to be converted to binary is at the center of this box and R is no more than half the image block length. The average of this box is calculated and if this average is less than the pixel value centered at $[2 \cdot R + 1]$-by-$[2 \cdot R + 1]$, then that pixel is set to 1, else 0. The blind average was again to high in this situation too. The average was reduce by 20%. Finally each block undergoes box counting. Box counting is recursive in nature and will use boxes to mark areas of interest. Below are the steps to compute the fractal dimension using the box counting method.

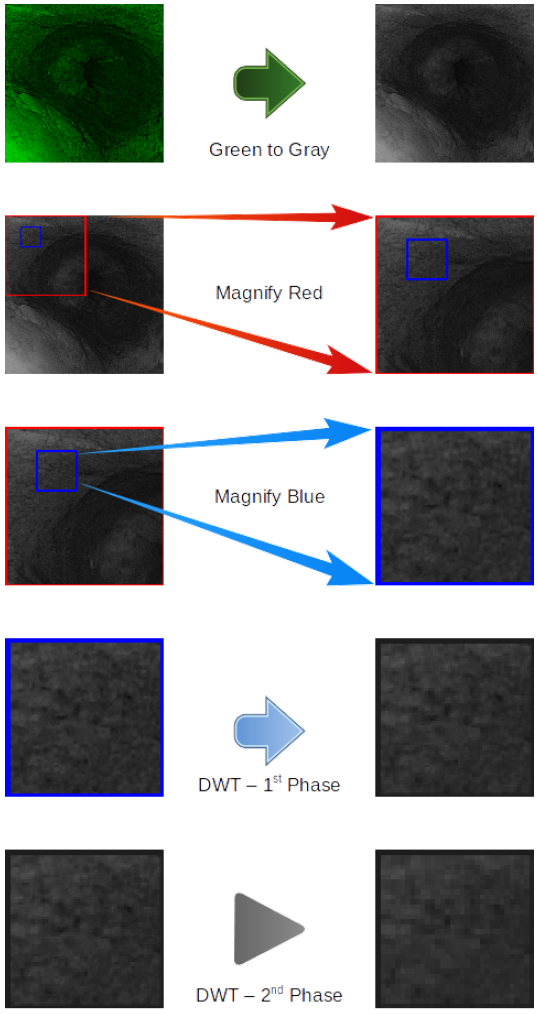


Figure 20: This is a demonstration of vein filtering done by DWT.

The implementation of box counting:

1) The image must converted to binary using dynamic thresholding.

2) Initial image length is L, then m is $log_2(L)$ or the number of box reductions (or number of loops).

3) The image is covered in enough boxes to cover the entire image. Round 1 will be one box, then 4 boxes in round 2, then 16 in round 3, etc.

4) After the size of the box has been determined, a count of all the boxes that contain a 1 inside that box is taken, and this is called n_i.

5) The Fractal Dimension (FD) is computed next, call the FD $D_i = \frac{log(n_i)}{log(d_i)}$, where d_i is the length of the box side at round i. 'i' begins at L and i is reduce by half at each round (L must be a power of 2).

6) Repeat steps 3-4, until i is 1.

7) Compute D_c, the fractal dimension for the entire image, and c is mapped to red (r), green (g), blue (b) or luminance (l). $D_c : \frac{1}{m} \cdot (D_1 + D_2 + ... + D_m)$.

8) Finally D is the product of all D_c, $D = D_r \cdot D_g \cdot D_b \cdot D_l$ is computed. This is the Fractal Dimension for the input image.

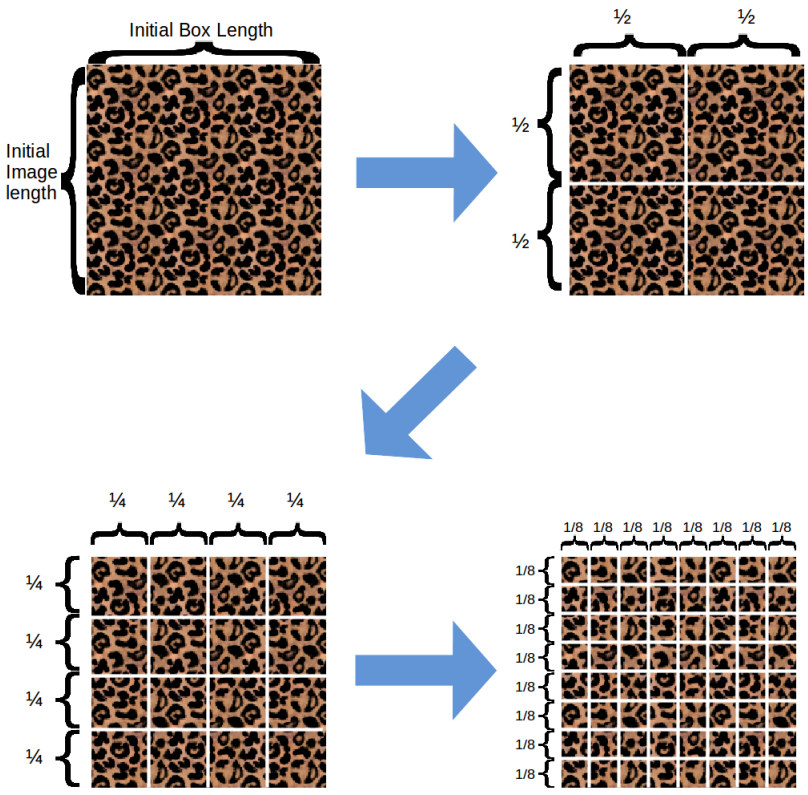


Figure 21: This is an example of the how box counting would work. All that needs to happen is to count the number of 1's in each box at each round or loop. This repeats until the box size is 1, then the results from each round/loop are summed together.

20	26	25	15	5	2	8	21	40	55	61	65	64	57	48	37
4	10	11	7	2	0	0	1	5	16	25	30	29	24	21	21
0	0	0	0	0	0	0	0	0	0	1	3	6	11	12	9
1	1	1	1	1	1	2	3	3	2	0	0	1	4	10	12
28	23	19	19	20	20	16	14	14	16	20	19	13	7	5	9
32	26	23	26	28	25	13	4	5	18	32	41	35	22	9	2
12	7	5	5	6	6	7	7	10	15	19	22	23	20	19	17
31	25	18	9	5	4	10	16	22	24	25	27	33	37	39	37
37	37	36	34	30	25	20	18	23	35	46	51	47	38	30	27
31	29	27	29	29	30	29	29	29	31	32	30	25	20	14	11
19	22	26	27	27	28	31	33	33	32	30	32	38	41	37	23
10	16	24	36	43	46	40	32	24	19	19	22	27	31	31	29
17	11	9	14	21	27	29	26	26	24	24	23	22	22	20	19
27	22	19	20	22	22	18	14	12	15	19	21	19	16	13	12
14	14	15	20	24	25	22	14	6	2	0	0	0	0	1	4
0	0	0	0	0	2	7	11	12	6	1	0	0	1	2	5

(a) before converting to binary

0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	1	1	1	1	1	0	0	0	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1
1	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0
0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0

(b) after converting to binary

Figure 22: this data was taken from a 16-by-16 pixel image.

4.6 Final Phase Notes

One reoccurring tool used in three of the four phases was the use of the Rect from the OpenCV libraries. The Rect is a template with very convenient functions that have also been optimized for GPUs when OpenCV is used. In phase 1, the Rect was used to focus a Region Of Interest containing the 1024-by-1024 subset of the input image. In phase 2, the Rect was used to apply the color intensity thresholding. In phase 4, the Rect was used to compute dynamic thresholding, and to reduce the input images to four smaller images as well as assisting in the box counting code. The Rect is a rectangle, in our proposed method all the Rects were set up as squares, same width and length. The heavy use of the Rect allowed the same code to be used for the sequential and the parallel version of our proposed method. All the image processing in the proposed method could be broken down to matrix operations, making the use of Rect a very smart, and lucky, choice.

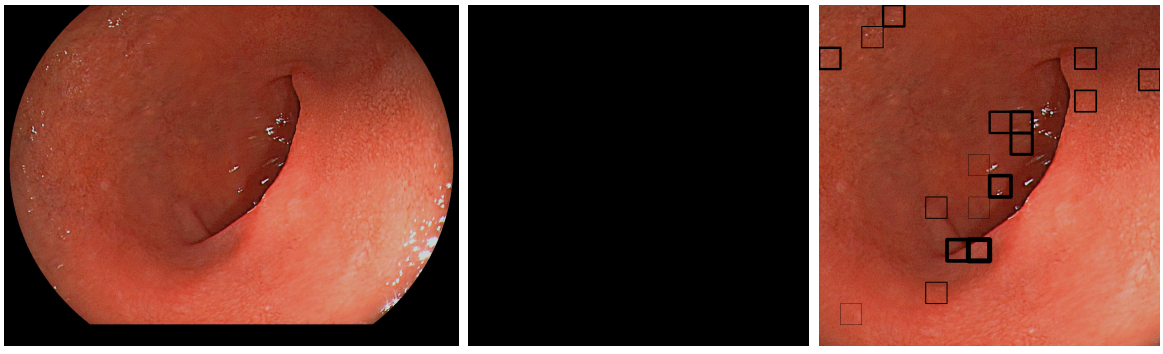
5 Results

The results are shown on a 1024-by-1024 pixel area used to detect early signs of esophageal cancer. The suspected areas will be contained in a black square box, 64-by-64 pixels in size. The edges of the squares are sensitive to the output of the Fractal Dimension (FD) results. This means that if the results of the FD computation are large, greater than 1.1, then the squares edge is very thick, if the results are small, less than 0.7, then the edges are very thin. This also means the thinner the squares edges, the greater the chances that the referred area may be an area of early cancer. Some images of patients that did not show signs of early cancer did trigger a square to be written to the image, meaning the proposed method detected a possible area of early esophageal cancer that was normal, or a false positive was detected. In testing, only three to ten squares trigger a false positive in the images without signs of early cancer, meaning there could be some simple math used to eliminate some of the false positives. One idea is to take the sum of all the reciprocals of all the Fractal Dimension results for an entire image and if the sum is not large enough, no boxes will be displayed.

5.1 Time Results

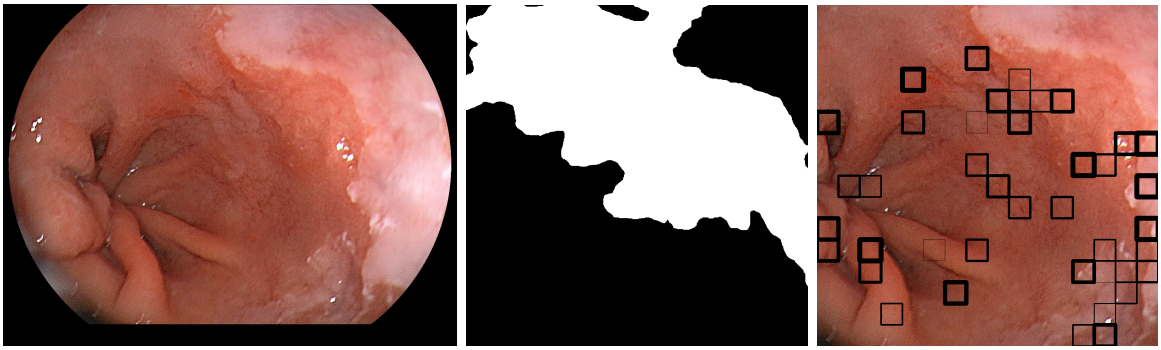
There were three testings platforms. The first was a Dell Studio XPS running Ubuntu 14.04 LTS, OpenCV 3.0 with an Intel Processor i7 @ 2.15 GHz and six GigaBytes (GB) of Random Access Memory (RAM). This setup used a NVIDIA 750i GPU running on the PCI bus with two GB DDR5 RAM. The next testing platform was a Macbook Pro with an Intel i7 processor @ 2.65 GHz and eight GB of RAM. This setup has an integrated Intel GPU, but this is not a powerful GPU, there is another GPU attached to this system for heavier graphics processing. The secondary GPU is an AMD GPU called a RadeonX 2000, it has two hundred and fifty six MegaBytes (MB) of RAM. The last testing platform is a Raspberry PI 2 with a quad core 900 MHz processor. This platform was chosen because is it very low cost and the GPU is built into the Central Processing Unit, like an Intel Iris processor. Additionally the CPU can access the GPU Level 2 Cache. The GPU memory for the Raspberry PI 2 was set to only 16 MB.

There were 3 images timed for the time tests. Each image was processed 100x and then the average time was calculated. The Raspberry PI only processed each image 25x, 100x cause instability, overheating and too many iterations could crash QT Creator. The best time came from the first platform with the NVIDIA GPU, less than two seconds per image. The second platform performed fairly well too, less than four seconds. There was no surprise the Raspberry PI 2 took longer than ten seconds but less than twenty seconds is still acceptable. See figure **??** for more time results.



(a) Pat. 1, Image 1, NBDT (b) Physician Annotation (c) Suspected Results

(d) Pat. 7, Image 1, ACHD (e) Physician Annotation (f) Suspected Results



(g) Pat. 10, Image 2, ACHD (h) Physician Annotation (i) Suspected Results

Figure 23: These images were provided to show that the output from a low cost system can still give correct results. These results were obtained from the Raspberry PI 2. See figure 24 for timing results.

The initial goal was ten seconds of processing per image, nearly 25x faster than Yamaguchi et al. The average results for a parallel implementation that includes saving and loading images to hard disc between each phase took only 1.5 seconds per image. This would mean that a one minute video at thirty frames per second would take forty-five minutes using the proposed method and it would take ninety hours for the work by Yamaguchi et al to process the same image.

Test Platform	Ave. Time Im1	Ave. Time Im3	Ave. Time Im3
PC + NVIDIA	1.4343 seconds	1.3937 seconds	1.3023 seconds
MAC + AMD	3.545 seconds	3.4964 seconds	3.4932 seconds
RASPBERRY PI 2	16.7267 seconds	16.2436 seconds	15.8685 seconds

Image 1	Patient 7 Image 2
Image 2	Patient 10 Image 2
Image 3	Patient 13 Image 2

Figure 24: This is a time table of results for 3 testing platforms.

5.2 Cancer Detection Results

The results were better than expected after thresholding was modified, see figure 14 for threshold modification general example. Before thresholding changed, most results were inconclusive. The output would show zero boxes marking early cancer or the whole image would be marked in boxes indicating early cancer. For example, when edge enhancement is applied by itself, all box counting results for every block fall in the range of 1.5 to 2.0. After all thresholding was applied, the results were more promising, all suspected areas of early cancer were identified. Using some elegant statistics the false positive rate could be reduced, however, a Support Vector Machine (SVM) would be better suited to identify the best thresholds and eliminate the false positives. Another goal for this research was to identify at least 50% of the areas from the physicians annotations. This goal was achieved when all areas of suspected early cancer using better thresholding and edge enhancement. The color components that gave the best results were green and blue, which also means luminance was producing fair results as well. The red color component contained values far too low and without much variation. The red color component was the worst for input data because of the red saturation of the endoscopic images. Other methods of enhancing edges may hold more promising results in the future but they need to be tested on a parallel platform.

All cases of early esophagus cancer annotated by an expert physicians were detected but the false positive rates were still greater than ideal. The positive results can be seen in the Suspected results images in figure 37. All areas annotated by a physician were detected, as can be seen in the results image in figure 23(c), there is a fair number of false positive hits. Figure 23(c) is an image that physician found had no signs of early cancer. While all areas of early cancer were identified, there was one set of results were not as good as figures 31(a)-31(c). The worst results came from figures 26(a)-26(c) The following results display images with identified early cancer only.

(a) Pat. 7, Image 2, ACHD (b) Physician Annotation (c) Suspected Results

Figure 25: Patient 7, Image 2.

(a) Pat. 7, Image 3, ACHD (b) Physician Annotation (c) Suspected Results

Figure 26: Patient 7, Image 3.

(a) Pat. 10, Image 1, ACHD (b) Physician Annotation (c) Suspected Results

Figure 27: Patient 10, Image 1.

(a) Pat. 10, Image 2, ACHD (b) Physician Annotation (c) Suspected Results

Figure 28: Patient 10, Image 2.

(a) Pat. 12, Image 2, ACHD (b) Physician Annotation (c) Suspected Results

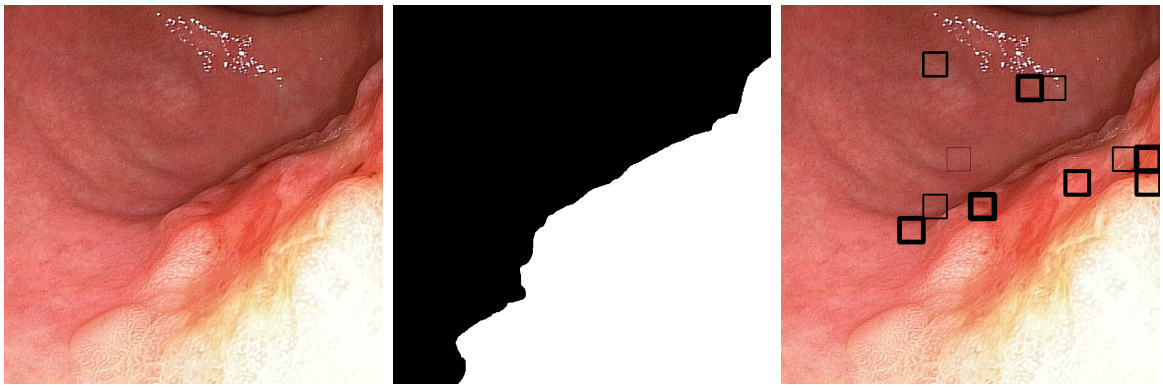
Figure 29: Patient 12, Image 2.

(a) Pat. 12, Image 3, ACHD (b) Physician Annotation (c) Suspected Results

Figure 30: Patient 12, Image 3.

(a) Pat. 13, Image 2, ACHD (b) Physician Annotation (c) Suspected Results

Figure 31: Patient 13, Image 2.



(a) Pat. 13, Image 4, ACHD (b) Physician Annotation (c) Suspected Results

Figure 32: Patient 13, Image 4.

(a) Pat. 16, Image 1, ACHD (b) Physician Annotation (c) Suspected Results

Figure 33: Patient 16, Image 1.

(a) Pat. 16, Image 3, ACHD (b) Physician Annotation (c) Suspected Results

Figure 34: Patient 16, Image 3.

(a) Pat. 17, Image 1, ACHD (b) Physician Annotation (c) Suspected Results

Figure 35: Patient 17, Image 1.

(a) Pat. 17, Image 5, ACHD (b) Physician Annotation (c) Suspected Results

Figure 36: Patient 17, Image 5.

(a) Pat. 19, Image 1, ACHD (b) Physician Annotation (c) Suspected Results

Figure 37: This is most likely the hardest area to detect. It was the smallest of the physician annotations, but there were results marked as areas of possible early cancer.

6 Conclusions

The proposed method was able to detect early cancer in every image provided. The results from the proposed methods showed some false positives, but the false positive rate dropped only a small amount when choosing thresholds with hand picked values. The Discrete Wavelet Transform phases successfully filtered the small veins and reduced the input space. Once the dynamic binary threshold was identified as being set too high when set to a strict average and reduced, the fractal dimension results increased in accuracy. From the results, annotated by a physician, early cancer signs were identified in every image containing cancer. But some of the images a physician concluded had no signs of early cancer were identified as containing signs of early cancer by the proposed method, producing false positives. This could be handled by a Support Vector Machine with the proper training. The goal of ten seconds or less was achieved with a NVIDIA and an AMD Graphics Processing Unit. The surprising results came from the Raspberry PI 2, this $35 board was able to process the image in less than twenty seconds and this time could be reduced with more RAM dedicated to the GPU.

7 Future Work

A Support Vector Machine (SVM) could be the optimal tool to zero in on better thresholds for color intensity, dynamic thresholding and fractal dimension results. This is possible

because a SVM would be able to use the thresholds set at a starting point, and the results of the proposed method would be a good start. The SVM is able to train itself from other images with a starting set of threshold values and the set better thresholds as it trains. Additional future work could include testing the proposed method using all color components except the red color component to evaluate the reduction in false positives, the red component could be dropped after the luminance calculations.

References

[1] *Cancer Facts & Figures 2016*, American Cancer Society Std., 2016. [Online]. Available: http://www.cancer.org/acs/groups/content/@research/documents/document/acspc-047079.pdf

[2] F. Hvid-Jensen, L. Pedersen, A. M. Drewes, H. T. Sørensen, and P. Funch-Jensen, "Incidence of adenocarcinoma among patients with barrett's esophagus," *New England Journal of Medicine*, vol. 365, no. 15, pp. 1375–1383, 2011, pMID: 21995385. [Online]. Available: http://dx.doi.org/10.1056/NEJMoa1103042

[3] *http://www.uhhospitals.org/health-and-wellness/health-library/a-c/barretts-esophagus#prettyP* Online Article, University Hospitals Std., 2016. [Online]. Available: http://www.uhhospitals.org/health-and-wellness/health-library/a-c/barretts-esophagus\#prettyPhoto

[4] S. K. Bram van Ginneken and C. Shneider., 2012-2016. [Online]. Available: http://endovissub-barrett.grand-challenge.org/

[5] O. D. Team. (2016, mar) Opencv documentation. [Online]. Available: http://docs.opencv.org/2.4/modules/core/doc/intro.html

[6] R. Y. Robert X Gao, *WAVELETS*. Springer Science+Business Media, 2011.

[7] D. Bryant. (2013, mar) Fourier transforms. [Online]. Available: http://devonbryant.github.io/images/fourier/fourier_transform.png

[8] R. POLIKAR, *THE ENGINEER'S ULTIMATE GUIDE TO WAVELET ANALYSIS*, Rowan University Std., mar 1999. [Online]. Available: http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html

[9] R. Polikar, "Wavelet tutorial," 2011-04-11 11:36:24. [Online]. Available: http://users.rowan.edu/\~{}polikar/wavelets/wttutorial.html

[10] J. J. O'Connor and E. F. Robertson, "Alfréd haar," *TURNBULL WWW SERVER*, aug 2006.

[11] W. Klonowski, "Signal and image analysis using chaos theory and fractal geometry," *Machine Graphics and Vision*, vol. 9, no. 1/2, pp. 403–432, 2000.

[12] D. Pearcy. (2013, may) Introduction to fractal geometry. [Online]. Available: https://danpearcymaths.files.wordpress.com/2013/05/image21.png

[13] J. Yamaguchi, A. Yoneyama, and T. Minamoto, "Automatic detection of early esophageal cancer from endoscope image using fractal dimension and discrete wavelet transform," in *Information Technology - New Generations (ITNG), 2015 12th International Conference on*, April 2015, pp. 317–322.

[14] D. Daniel. (2011, dec) Fractal dimension. [Online]. Available: http://soft-matter.seas.harvard.edu/index.php/Fractal_Dimension

[15] Itseez. (2016, mar) Cuda. [Online]. Available: http://opencv.org/wp-content/uploads/2012/06/perf.png

[16] ——. (2016) Opencv. [Online]. Available: http://itseez.com/opencv/

[17] I. Corporation. (2016) hd-graphics-developer. [Online]. Available: http://www.intel.com/content/www/us/en/architecture-and-technology/hd-graphics/hd-graphics-developer.html

[18] N. Corporation. (2016) What is cuda? [Online]. Available: http://www.nvidia.com/object/cuda_home_new.html

[19] M. Wolfe. (2012, dec) Understanding the cuda data parallel threading model a primer. [Online]. Available: https://www.pgroup.com/lit/articles/insider/v2n1a5.htm

[20] N. Ralph, "Lab tested: Amd's bulldozer packs plenty of cores, but not enough power," Online Article, 2016.

[21] F. van der Sommen, S. Zinger, E. Schoon, and P. de With, "Supportive automatic annotation of early esophageal cancer using local gabor and color features," *Neurocomputing*, vol. 144, pp. 92–106, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231214007401

[22] G.-H. K. Kwang-Baek KIM, Sungshin KIM, "Analysis system of endoscopic image of early gastric cancer," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E89-A, no. 10, pp. 2662–2669, oct 2006.

[23] D. Nexus. (2012, dec) Gorgeous hair and amazing eyes. [Online]. Available: https://people.desktopnexus.com/wallpaper/1334970/

VITA

Author: Jason Helms

Place of Birth: Spokane, Washington

Undergraduate Schools Attended: Eastern Washington University

Degrees Awarded: Masters of Computer Science, 2016, Eastern Washington University

Bachelor of Computer Science, 2013, Eastern Washington University

Honors and Awards: Graduate Assistantship, Computer Science Department
2013 – 2015, Eastern Washington University

EWU Linux Users Group President 2014 – 2015

EWU Linux Users Group Vice President 2013 – 2014

Professional Experience: Graduate Assistant, System Administrator, Eastern Washington University, Cheney, Washington, Jan. 2013 – Dec. 2015

MANOME
(Metropolitan Area Network Optimized Musical Environment)
Single Board Computer (SBC) researcher
University, Cheney, Washington, Jan. 2013 – June. 2015

Work Study, Spokane Falls Community College, Spokane, Washington, Sep. 2011 – Dec. 2012